

# Make Claude Yours — Memory, Custom Commands & Automation

The difference between using AI and owning a system: Claude that remembers your business, runs your playbooks, and works while you don't.

**stuck.builders**

Build systems, not tabs

## CLAUDE.MD — GIVE CLAUDE A PERMANENT MEMORY

| FILE                        | WHAT IT'S FOR                                       |
|-----------------------------|-----------------------------------------------------|
| ~/ <b>.claude/CLAUDE.md</b> | You, everywhere — your preferences in every project |
| <b>./CLAUDE.md</b>          | The project — team rules, shipped via git           |
| <b>./CLAUDE.local.md</b>    | You, this project only — never committed            |

Claude loads these automatically at launch, broadest first. Put in them: what your business does, how you like things written, standing rules ("always confirm before sending anything external"). Run **/init** once to generate a starter, **/memory** to edit anytime. Reference other docs inline with **@** — e.g. "See *@docs/offer.md* for pricing" — and they load too.

## HEADLESS MODE — CLAUDE IN YOUR SCRIPTS

```
# Ask and exit — no interactive session
claude -p "summarize today's sales log"

# Pipe anything in
cat leads.csv | claude -p "top 10 by fit, as a table"

# Structured output for automations
claude -p "..." --output-format json

# Continue where you left off
claude -c
```

This is the bridge from "chatting with AI" to "AI inside your operations" — cron jobs, daily digests, report generation, data cleanup. If it runs on a schedule, it can run through Claude.

## CUSTOM COMMANDS — TURN PLAYBOOKS INTO /COMMANDS

Any markdown file in **.claude/skills/** (or **~/claude/skills/** for personal, global ones) becomes a slash command. Write the playbook once, run it forever:

```
# File: .claude/skills/followup/SKILL.md
---
description: Draft my client follow-up email
argument-hint: [client-name]
---
Draft a follow-up email to $0 in my voice: warm, short, one clear next step.
Check @clients/$0.md for context first.
```

Now **/followup acme** runs the whole play. **\$0**, **\$1...** are arguments; **\$ARGUMENTS** grabs them all. Every repeatable task you do is a candidate: proposals, reports, onboarding, weekly reviews.

## CONNECT YOUR TOOLS (MCP)

```
# See what's connected
claude mcp list

# Manage connections in-session
/mcp
```

MCP servers plug Claude into Gmail, HubSpot, Stripe, Slack, your database — so it acts on real business data instead of copy-paste. This is the "operating system" layer.

**The 4-step ownership ladder:** ① CLAUDE.md so it knows your business → ② custom /commands for your repeatable plays → ③ MCP so it touches your real tools → ④ headless mode so it runs without you. Most people never get past step 0. That's the whole gap.